

TRGSim++ status

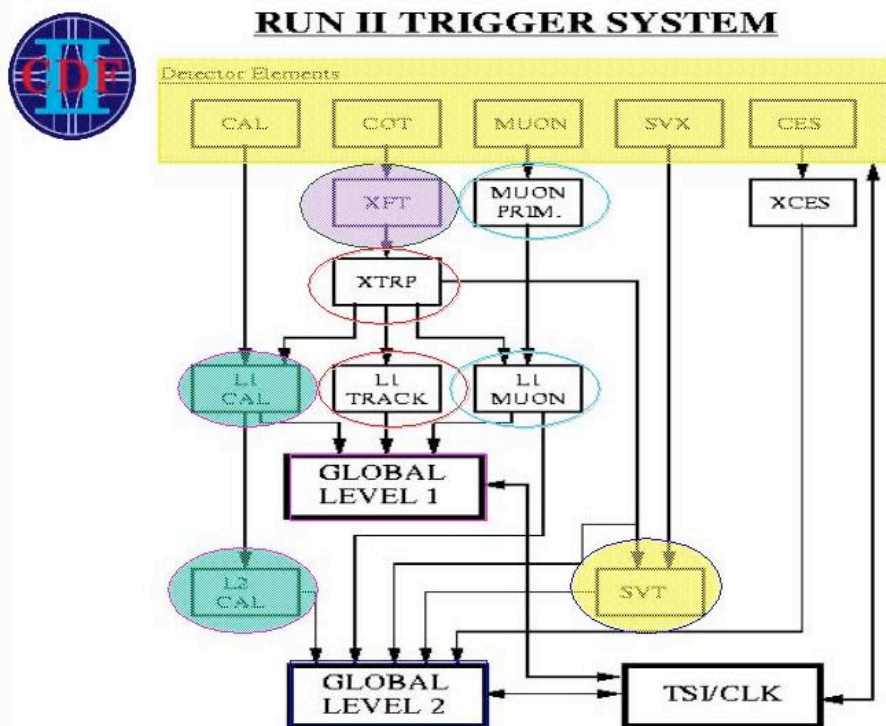


TRGSim++ is a set of (C++) packages emulating the various trigger levels decision steps

Trigger decision steps: A_C++ modules, organized in packages:

CalTrigger -> CalTriggerExe
MuonTrigger -> MuonTriggerExe
XFTSim -> XFTTest (tbin)
svtsim -> svtsimtest (tbin)
XTRPSim -> XTRPSimExe(tbin)
L2GlobalTrigger -> L2Sim
L1GlobalTrigger -> FredSim
TriggerMods -> TRGSim++
TriggerObjects -> trigger banks

TRGSim++ modules run off detector raw data and produce emulated trigger data identical to real hardware data.
(It also runs off COTQ, CalData, PadSqr::COTQ and PadSqr::SVXQ)



Documentation and examples



<http://ncdf70.fnal.gov:8001/trigsim/trgsim.html>

accessible from the
CDF FastNavigator

CDF Fast Navigator [by Martin Hennecke](#)

CDF Home	Online				
Shift tools	Beam status	MCR e-log	Shift e-log	e-logs	Run summary Expert phone list
	DAQ Ace	Mon Ace	CO	SciCo	Shift schedules
Operation	B0 Home	CDF Weekly Meeting	Ops Manager Weekly Plan		CDF Safety Training records
Detector	Detector Groups				Upgrades
	Silicon COT	Calorimeter Muon	CLC TOF	Forward Detectors	Run IIa / Run IIb Run IIb Task Force
Trigger	Trigger Home	L2 / L3	Trigger WG	B Trigger	Exotics Trigger
Data quality	DQM Home	Goodrun lists	Consumer Home		

	Computing				
Software	Code management Validation	Codebrowser DB Browser Calib DB	MC production	Stntuple / evtNtuple TopEventModule TopNtuple / PAX	cdfSim / TrigSim Event Display CR Tagger
	CAF Home Active CAF queues	SAM / Grid Grid mails	Data Handling Enstore	CDF System Status	Printers
Hardware					

Documentation and examples (II)



RUN II TRIGGER SYSTEM

The Trigger Simulation Project

Project Leader: [Simona Rolli](#)

mail to: [TRGSim team](#)

[offline](#) [online](#) and [TrigMon](#)

- [Release tags and notes \(formerly List of changes and bug fixes\)](#)
- [TCL file to run TRGSim++ on real data](#) (September 2006, 6.1.4 release)
- [TCL file to run TRGSim++ on MC data](#) (September 2006, 6.1.4 release)
- [Dan's validation page](#): TRGSim++ is run and checked for every release.
- [CalTrigger](#)
 - [The L1Calorimeter triggers -DIRAC](#)
 - [DCAS Trigger Map](#)
- [XFTSim](#)
- [XTRPSim](#)
- [MuonTrigger](#)

To run TRGSim++ look at the tcl's on the web page or in the release TriggerMods/test area

They are also available from the head of the repository

http://cdfcodebrowser.fnal.gov/CdfCode/source/TriggerMods/test/run_TRGSim++.tcl

http://cdfcodebrowser.fnal.gov/CdfCode/source/TriggerMods/test/run_TRGSim++_MC.tcl

Releases status -web page



<http://ncdf70.fnal.gov:8001/trigsim/TACS.html>

This file reports the latest tags to fix various bugs discovered during use and to add new features.

When a release is not mentioned it means that the features of **TRGSim++** are of nearest previously mentioned release.

For the releases status look [here](#)

TRGSim++ 6.1.4 (Jan 13, 2006)

- no changes
- when running on MC simulated data, `prereq::_printSummary()` method will not report the correct number of events passing L2 triggers. It will report instead that no events pass any L2 trigger. This is bug in `prereq` that was reported by Simona and confirmed by other users. She has a modified version of `Prereq.cc` which prints the correct number of events passing the L2 triggers.

TRGSim++ 6.1.3 (Sept 20, 2005)

- **CalTrigger 6.1.3**
Peter modified several parts of CalTrigger to simplify/add to the diagnostic needed in the Control room, when CalTrigger is run as part of TrigMon.
- **L2GlobalTrigger**

TRGSim++ 6.1.2 (July 27, 2005)

- no changes

TRGSim++ 6.1.1 (Apr 2005)

- **CalTrigger v6_1_1pre1**
added correct fetching of DIRAC granularity, had/em parameters from db, as requested by Peter Wilson (needed online) added fetching of spikekiller parameter as requested by Peter Wilson.
- **TriggerObjects v6_1_1pre1**
- **BankTools v6_1_1pre1**
David Dagenhart repositied a new bank class for the XFT system, XTCC.
- **L2GlobalTrigger and TriggerDB v6_1_1pre1**
Changes to TriggerDB are to add recognition of L1 FORCE SVT for svtsim, and removing an annoying printout that was cluttering TrigSim log

Release Status



- **TRGSim++ 6.1.4 (Jan 13, 2006)**

when running on MC simulated data, `prereq::_printSummary()` method will not report the correct number of events passing L2 triggers. It will report instead that no events pass any L2 trigger. This is bug in prereq that was reported and confirmed by other users.
Private modified version of Prereq.cc which prints the correct number of events passing the L2 triggers exists.

- **TRGSim++ 6.1.3 (Sept 20, 2005)**

CalTrigger : Peter W. modified several parts of CalTrigger to simplify/add to the diagnostic needed in the Control room, when CalTrigger is run as part of TrigMon.
L2GlobalTrigger had some changes to adequate to new trigger tables

- **TRGSim++ 6.1.1 (Apr 2005)**

CalTrigger: added correct fetching of DIRAC granularity, had/em parameters and spikekiller thresholds from db
TriggerObjects, BankTools: new bank class for the XFT system, XTCC.
L2GlobalTrigger and TriggerDB: changes to TriggerDB to add recognition of L1_FORCE_SVT for svtsim, For L2GlobalTrigger few updates to match the online L2 algorithm codes.
XTRPSim - minor changes
svtsim - various changes - some related to DB



Release status (cont'd)

- **TRGSim++ 6.0.0 (Dec 2004)**

XTRPSim: new code for the three-tracks trigger board.

XFTSim: new firmware code.

L2GlobalTrigger: some updates to the L2 decision simulation to handle muon triggers and the SVT delta-z cuts, among other things.

MuonTrigger: extensive changes to introduce forward muon trigger primitives and triggers

CalTrigger: tagged to make sure that the SCL for MC are picked up correctly

- **TRGSim++ 5.3.1 (January 2004)**

it includes Prereq to select events in output based on the trigger bits

it includes cotqModule,svqModule,calqModule and sxmqModule to unpack Pad squeezed banks in input

- **TRGSim++ 5.3.0 (Fall 2003)**

it does not include Prereq to select events in output based on the trigger bits.

Corrections were added for the IMU maps in XTRPSim

Release status (cont'd)



- TRGSim++ 5.2.0int2(Fall 2002)

it includes infrastructure changes to improve the database connection and a fix for the Sumet thresholds used in the CalTrigger Prefred simulation - the sumet bits are correctly set.

Starting with 5.2.0int2 XFTSim should not produce random crashes at the end of job, as a major rewriting of the code has been made by Ben Whitehouse.

- TRGSim++ 4.10.x and 4.11.1 (Spring 2003)

it includes a change in the FRED bits scheme. Incorrect Sumet and MET thresholds, values are correct, bits are incorrect.

- ♦ if you want to correctly emulate the Sumet bits with this release (not Sumet value!) add the package CalTrigger : [addpkg CalTrigger PreFred_091103](#)
- ♦ better to use 5.x releases!

- TRGSim++ 4.9.x (Fall 2002)

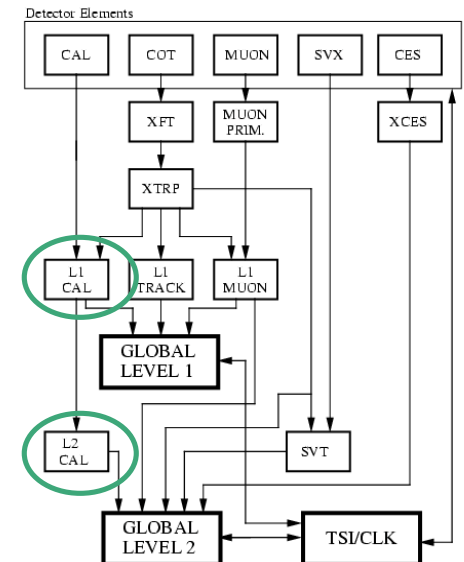
it includes modifications to the db connection (tcl changes to CalibrationManager talk-to from 4.9.1 to 4.9.1pms)

Code: CalTrigger



- Trigger Tower energies (TC2D – first word)
 - ♦ <http://ncdf70.fnal.gov:8001/trigsim/DCAS.html>
- L1 DIRAC Triggers (TC1D)
 - ♦ http://ncdf70.fnal.gov:8001/trigsim/dirac_trigger.html
- L2 clustering and Iso sums (TC2D – second word)
 - ♦ L2 clustering will change, following update specs (cdfnote 8415)
- database access for trigger definition - real data
 - ♦ possibility to run on simulated run with conditions from real run: the value is fetched from the event record or in talk-to

```
use_software_CAL_banks set t
use_xtrp set t
use_hardware_xtrd set f
use_hardware_L1 set f
run_Number set 151435
```



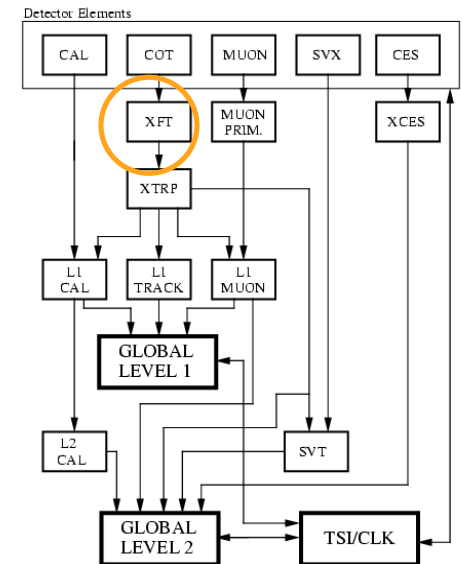
CalTrigger input/output

- Calorimeter D-banks: CEMD, CHAD, PEMD, PHAD, WHAD.
 - ♦ It alternatively reads CalData and puffs it into D-banks, via a call to CalorimetryModule.
- Track input from XTRD_StorableBank.
 - ♦ 2 options to get the track input controlled by the parameter "standalone" in the talk-to:
 - from real data (standalone = 1)
 - from a simulated XTRD (standalone = 0).
- Output :
 - ♦ **TC2D** with the list of trigger tower energies and clustering summary word for each tower, description "**Simulated Trigger Bank**";
 - ♦ **TC1D** with the results of the DIRAC triggers: description "**Simulated Trigger Bank**" ;
 - ♦ **TL1D** with the PreFred bits for CalTrigger and SumEt: description "**CalTrig_Simulated_Trigger_Bank**", it used later by TL1D Maker and dropped in output
 - ♦ **TL2D** where the only block filled is the clustering block: description "**Calorimeter_TL2D_Bank**", used by L2Sim later and dropped in output

Code: XFTSim



- XFT tracks :
 - ◆ XFLD and XFFD diagnostic banks
 - ◆ XTRD (from XTRPSim) includes the tracks parameters
- DB access implemented
 - ◆ real run conditions applied to simulated run:
- For the Road and Mask files one can set the file types. So one can mimic exactly what was run.



XFT input/output

- **Input:**
 - ◆ **COTD hit information** (void XFTSim::getInputData())
 - ◆ COTQ if COTD is not found in data stream
 - ◆ 5.x releases requires **FOR MC** that after loading the COT hits, one should add the pedestal of 132 ns.
 - ◆ 5.3.1 release require the use of cotqModule to unpuff PadSqz::COTQ
- **Output:**
 - ◆ XFFD - XFT Finder bank
 - ◆ XFLD - XFT Linker bank

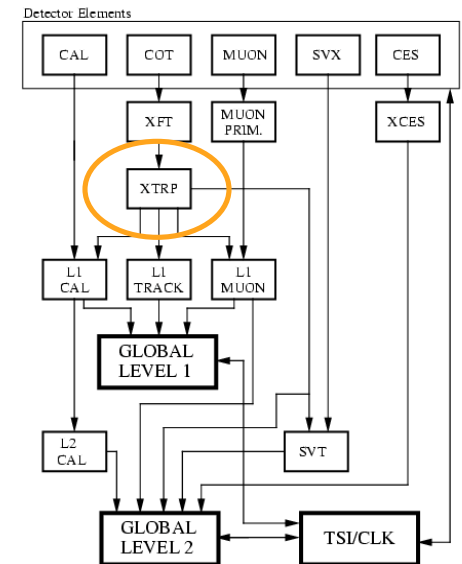
} Input to XTRPSim

Code : XTRPSim



XTRP tracks:

- ◆ XTRD
- database access for trigger definition - real data
 - ◆ TriggerDB DOWNLOADS Table : XFT_PT
 - ◆ possibility to run on simulated run with conditions from real run: the value is fetched from event record or in talk-to standalone set f
run_Number set 151435
- It is possible to set the muon pt thresholds by hand, overwriting the database values.



Code : XTRPSim

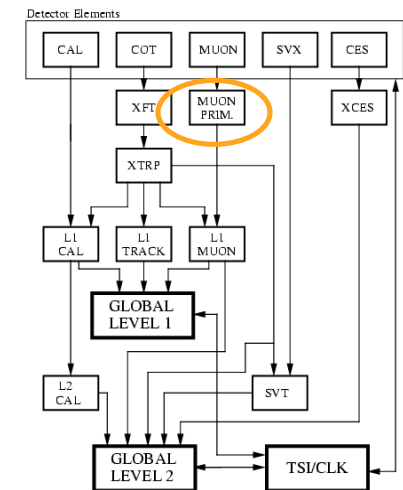
- Input : XFLD_StorableBank
 - ♦ real or simulated controlled via talk-to parameter "*standalone*"
- output XTRD_StorableBank:
 - ♦ track data: 2 blocks:
 - first block: 12 cards (2 wedges per card)
 - Track data:
 - ϕ
 - p_T
 - Isolation bit
 - short track bit
 - XTRP/Calorimeter bus
 - 2nd block: track trigger bits word

Code: MuonTrigger



- It emulates L1 muon primitives:
 - ♦ Xtrapolation with XTRP done
 - ♦ stubs thresholds hardwired (no db access)

Detector	Basic Unit	Unit	CARD	Algorithm Description	# Outputs
CMU	Wire pair	1.25°	MUIT	Hi, low P_t , plus a "lefover to 384ns" determined from differential timing.	288 x 2 x 2 $\phi \pm \eta \pm P_t$
ChP	4 tube stack	0.6° 1.2°	MPIT	2 or 3 out of 4 hits for patterns from radial tracks	336 ϕ
CMX	Wire pair	1.25°	MXIT	Hi, low P_t , plus a "lefover to 384ns" determined from differential timing	288 x 2 x 2 $\phi \pm \eta \pm P_t$
CSX	Coincidence	15°/8	MSIX	Coted Mean Time from 1/2 overlapped scintillators	192 x 2 $\phi \pm \eta$
CSP	Scintillator	1.2° 2.4°	MSIP	Coted Scintillator hit	168 x 2 $\phi \pm \eta$
HAD	Calorimeter	15°	MHIT	Signal in calorimeter PMT for η intervals 0/4, 4/5 and 6/9	24 x 6 $\phi \pm \eta$



MuonPrimitive0

MuonPrimitive1

MuonPrimitive2

MuonPrimitive3

MuonPrimitive4

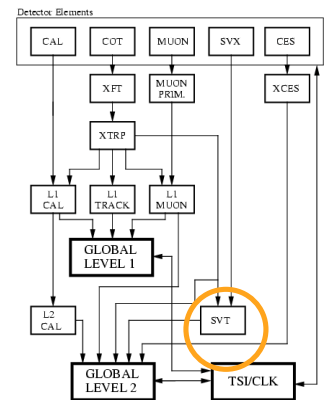
MuonPrimitive5

MuonTrigger input/output

- **Input:**
 - ♦ CMUD_StorableBank
 - ♦ CMXD_StorableBank
 - ♦ CMPD_StorableBank
 - ♦ XTRPMatchBoxData
 - ♦ BMUD_StorableBank
 - ♦ BSUD_StorableBank
 - ♦ TSUD_StorableBank
 - ♦ HATD_StorableBank
- **Output:**
 - ♦ TCMD_StorableBank

Code: svtsim

- It produces SVT tracks, it does not emulate the L2 decision;
- No private version, the one in the repository is the only general use version;
- Ability to run MC realistic simulation with SVT beamlines.



```

module talk svtsim
  dataFileDir set "$env(CDFSOF2_DIR)/svtsim/svtdata"
  writeBank set true
  useDB set true
  debugPrint set f
  sixdProcessName set "NSIM"
  sixdDescription set "CORRECTED"
  BeamMenu
    useBeamLineFromDatabase set t
    beamDatabaseFileName set "$env(CDFSOF2_DIR)/svtsim/test/testSVTBeam"
  exit
exit
  
```

Input
SIXD and XFLD

-
- ```

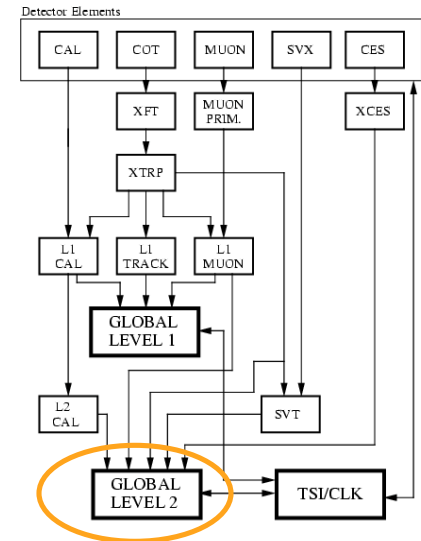
graph TD
 subgraph Detector_Elements [Detector Elements]
 CAL
 COT
 MUON
 SVX
 CES
 end
 CAL --> XFT
 MUON --> MUON_PRIM[MUON PRIM.]
 CES --> XCES
 XFT --> XTRP
 MUON_PRIM --> XTRP
 XCES --> XTRP
 XTRP --> L1_CAL[L1 CAL]
 XTRP --> L1_TRACK[L1 TRACK]
 XTRP --> L1_MUON[L1 MUON]
 L1_CAL --> GLOBAL_LEVEL_1[GLOBAL LEVEL 1]
 L1_TRACK --> GLOBAL_LEVEL_1
 L1_MUON --> GLOBAL_LEVEL_1
 GLOBAL_LEVEL_1 --> GLOBAL_LEVEL_2[GLOBAL LEVEL 2]
 SVX --> SVT
 SVT --> GLOBAL_LEVEL_2
 CES --> TSI_CLK[TSI/CLK]
 TSI_CLK --> GLOBAL_LEVEL_2

```

# Code: L2GlobalTrigger



- L2Sim: L2 bits + L2 quantities
  - TL2D simulated includes:
    - L1 Trigger Card: L1 bits
    - L2 Trigger Card: L2 bits
    - Reces - XCES Trigger Card
    - SVT - SVT Card
    - Cluster Card
    - Isolation Cluster Card
    - XTRP Card ( match word not filled)
    - Muon pulsar
- ◆ L2 triggers emulated, after getting their definition from TriggerDB
- ◆ Some things are not up-to-date: dimuon mass cut
  - Not clear that L2GlobalTrigger authors will do it



# Code: TriggerMods

TriggerMods is the  
placeholder  
package for the  
global executable,  
**TRGSim++**

```
AppUserBuild::AppUserBuild(AppFramework* theFramework)
: AppBuild(theFramework)
{
 // framework
 addCDFrequiredModules(this);
 addAllStorableObjects(); // production / physics objects
 addMiscStorableObjects(); // everything else
 addXTRPSim(); // XTRPMatchData
 // dump
 add(new EventDump("EventDump", "EventDump"));
 // several inputs
 add(new YbosDiskFileInputModule("YbosDiskFileInput", "Input Module for TRYBOS files"));
 add(new XXXDiskFileInputModule("XXXDiskFileInput", "Input Module for XXX files"));
 add(new APPConsumerInputModule("ConsumerInput", "Consumer Input Module"));
 // Puffing modules
 add (new CalqModule());
 add (new SmxqModule());
 add (new CotqModule());
 add (new SvqxModule());
 // Trigger emulation modules
 add(new CalorimetryModule ("CalorimetryModule", "CalModule "));
 add(new XFTSim("XFTSim", "XFT trigger simulation Module"));
 add(new XTRPSimModule("XTRPSim", "XTRP trigger simulation Module"));
 add(new svtsimmodule("svtsim", "SVT trigger simulation Module"));
 add(new calor::CalTriggerDataMaker("CalTriggerDataMaker", "Calorimeter trigger simulation Module"));
 add(new muon::MuonTriggerMaker("MuonTriggerMaker", "Muon trigger simulation Module"));
 add(new TL1DMaker("TL1DMaker", "TL1D simulation module"));
 add(new L1GlobalTriggerMaker("FredSim", "Fred trigger simulation Module"));
 add(new SpikeFilter("SpikeFilter", " Filter for trigger towers energy below th"));
 add(new L2SimModule("L2Sim", "L2 trigger simulation Module"));

 // monitoring code - necessary for data
 add(new NewTrigSimModule("TrigSimModule", "TrigSim for a consumer program"));

 // Prereq filter
 add(new Prereq());
}
```

# Trigger banks

---

- **TriggerObjects**
  - ◆ trigger banks as in [cdf\\_note4152](#)
  - ◆ simple accessors ( `get_word ( int, int, int)` )
  - ◆ named accessors ( depending on the bank )
    - look at the code browser
  - ◆ Examples of accessors: `TriggerObjects/ntuple`



# Conclusions

---

- TRGSim++ has been used since October 2000 (commissioning run!)
- In its present form and functionality it provides emulation of all trigger steps since Summer 2002 (when the last piece was added, L2Sim)
- It is routinely built as part of all the integration releases and frozen releases.
- Pretty much up-to-date as for upgraded trigger systems
- Part of it runs in control room as part of TrigMon

<http://ncdf70.fnal.gov:8001/trigsim/trigsim.html>